**统一会话管理共享平台**

## 1、摘要

统一会话管理共享平台基于开源j2ee框架bboss开发，遵循j2ee servlet 2/3规范，使用mongodb存储会话信息，使用bboss序列化机制序列化会话数据，以特有的方式解决了会话共享五大技术难题：

1. session数据序列化问题，
2. session sticking问题，
3. 跨域跨应用session共享问题，
4. 跨容器(tomcat,jetty,weblogic，websphere等) session共享问题。
5. SSO单点登录单点登出一致性问题

## 2、会话共享现状

目前，业界实现session共享主要有如下两种方案：

1. 应用服务器间的session复制共享（如tomcat session共享）
2. 基于cache DB缓存的session共享（如redis,memcached）

**基于容器提供的session复制共享方案有很多不足：**

1. 技术复杂,必须在同一种中间件之间完成(如:tomcat-tomcat之间).
2. session复制带来的性能损失会快速增加.特别是当session中保存了较大的对象,而且对象变化较快时, 性能下降更加显著. 这种特性使得web应用的水平扩展受到了限制。
3. Session内容序列化（serialize），要求对象类都实现java.io.Serializable接口，并且会消耗系统性能。
4. Session内容通过广播同步给成员，会造成网络流量瓶颈，即便是内网也会存在瓶颈。

基于cache DB缓存session共享方案将session数据统一存储到cache db中，避免了各应用节点之间的数据广播和复制，虽然在很大程度上解决了容器session复制共享方案中存在的问题，并且有基于memcached和redis的开源实现，但是这些实现也往往是基于特定的容器实现的（如tomcat），不能做到跨容器的会话共享，cache db服务器故障时，可能会导致数据丢失，基本上没有提供统一管理和监控session数据以及在线用户数统计功能。

统一会话管理共享平台使用mongodb来结构化存储会话数据，很好地解决了上述两种方案的不足，为应用系统提供了非常廉价的开源会话共享解决方案。

## 3、统一会话共享作用和特点

**作用：**为应用提供统一会话管理功能，避免集群部署场景下负载切换session丢失问题；跨域跨应用会话共享并实现SSO功能（统一登录统一登出系统）。

**存储：**采用mongodb存储会话数据，采用增量模式修改会话属性，读写分离，简单高效;应用可以对会话数据进行请求级缓存（一次请求处理过程中，同样的属性只获取一次，第二次就在本地内存获取，避免多次获取相同属性不必要的开销，请求结束后立即清除本地缓存数据，避免脏读）。

**序列化：**采用bboss序列化机制以xml格式序列化会话数据，可读性好，易于监控，提供序列化插件，扩展性强

**规范：**遵循servlet 2/3规范，可无缝与现有应用系统集成，应用无需或者只需少量修改代码。Session监听器需遵循bboss会话共享规范，需将原来容器 session监听器迁移到bboss会话共享实现。如修改session中对象数据，必须调用session.setAttribute方法将对象数据更新到mongodb中，以便将更新后的数据共享给其他应用。

**兼容性：**跨容器，兼容业界主流的应用服务器（tomcat,weblogic,webspere,jetty）），支持容器会话管理和bboss会话管理两种机制，可根据实际需要自由切换应用会话管理机制。

**约束：**无约束，无需session sticking，访问请求可以平均分派给各集群节点，支持lvs,haproxy，ngix 4,7层负载。

**安全性：**

客户端基于cookie机制存储sessionid，通过设置cookie **httponly**属性阻止XSS窃取sessionid，通过设置**secure**属性并结合https阻止传输过程中sessionid被窃取

**监管：**session信息统计查询, 应用在线用户数统计查询，应用会话管理功能（删除会话、 查看会话属性数据）

## 4、统一会话共享架构设计

1. 会话共享整体架构：

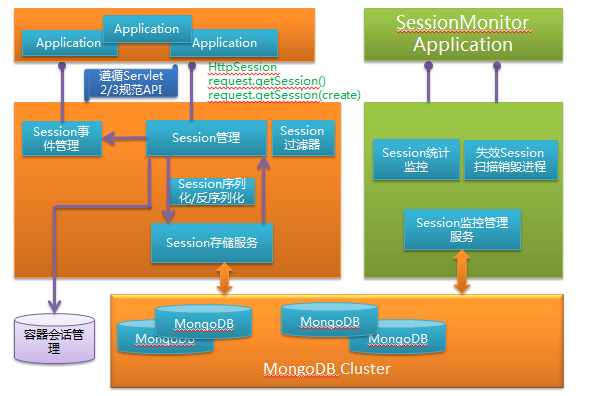
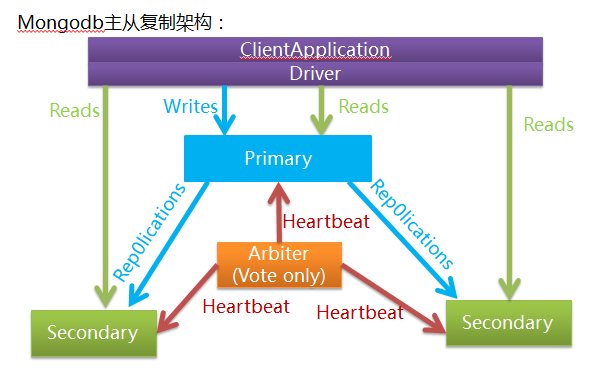


图1.统一会话管理平台体系架构

1. 高可用mongodb服务器部署架构：



1. Session对象在mongodb中的存储结构示意图：单应用集群节点之间共享会话存储结构。

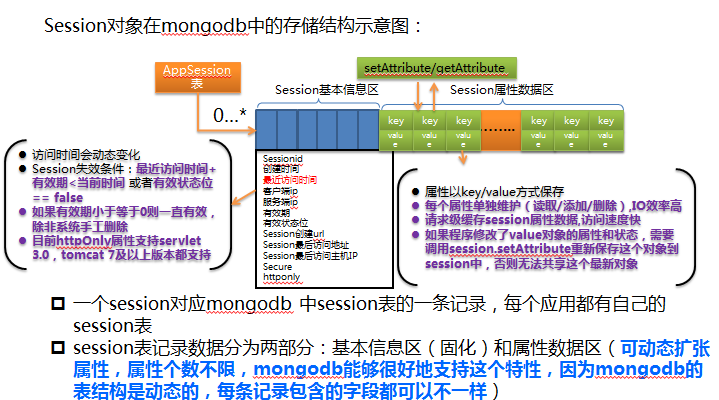
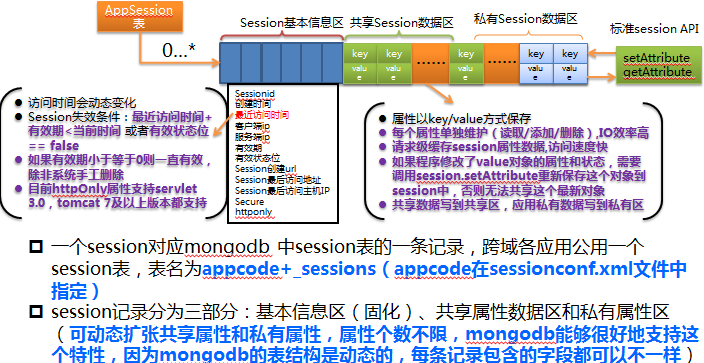


图2.统一会话管理平台mongodb存储结构设计

* 一个session对应mongodb 中session表的一条记录，每个应用都有自己的session表
* session表记录数据分为两部分：基本信息区（固化）和属性数据区（可动态扩张属性，属性个数不限，mongodb能够很好地支持这个特性，因为mongodb的表结构是动态的，每条记录包含的字段都可以不一样）

1. Session对象在mongodb中的存储结构示意图：跨域不同应用之间共享会话存储结构



* 一个session对应mongodb 中session表的一条记录，跨域各应用公用一个session表，表名为appcode+\_sessions（appcode在sessionconf.xml文件中指定）
* session记录分为三部分：基本信息区（固化）、共享属性数据区和私有属性区（可动态扩张共享属性和私有属性，属性个数不限，mongodb能够很好地支持这个特性，因为mongodb的表结构是动态的，每条记录包含的字段都可以不一样）

1. 会话共享-典型部署架构-双主备模式，如图所示，Session共享组件部署分别部署在4台应用服务器中：



## 5、统一会话共享模式

统一会话管理共享平台提供两种会话共享模式 ：

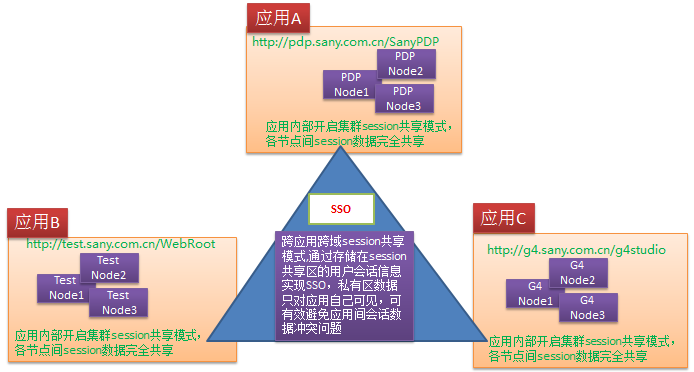
**模式一：** **集群间会话共享模式，**实现同一个应用集群节点之间的会话共享，通过这种模式可以避免因故障导致访问请求切换服务器时session丢失问题，同时也可以将同一用户访问请求平均分派到各个服务器上，达到真正的负载均衡。

**模式二：** **跨域跨应用模式**，实现同一域名或者同一根域（不同的子域名）下不同应用之间的会话共享，实现他们之间的单点登录功能（SSO）。

第一种模式相对简单，适用于应用集群间会话共享；第二种模式在配置方面比模式一稍微复杂一些，模式二可以灵活定义哪些会话数据需要在应用之间进行共享，哪些数据作为应用私有会话数据不对其他应用共享（这个在实际情况下很有用），默认情况下共享应用间的所有会话数据。

实际的应用环境中，模式一和模式二经常组合一起使用，每个应用本身采用集群部署模式（开启集群间会话共享模式），同时利用跨域跨应用模式实现不同应用间的单点登录功能（**前提是这些应用必须使用同一个域名或者都拥有相同的根域名，根域名不同的话可以使用bboss统一令牌系统实现跨根域系统之间的SSO**）。

两种会话共享模式一起使用示意图：



说明：图中三个应用，访问地址分别为

http://pdp.sany.com.cn:8080/SanyPDP

http://g4.sany.com.cn:169/g4studio

http://test.sany.com.cn:8080/WebRoot

三个应用的用户会话信息存储在session的共享属性区，这样可以实现三个应用之间的会话共享和单点登录功能。如果需要共享更多的会话数据，可以将对应的属性追加应用共享会话属性，其他属性都是私有会话数据（对其他应用不可见）。如果没有配置共享属性，那么所有会话数据全部在三个应用间共享。

## 6、性能测试

针对测试环境登录和登出场景进行测试，分别测试不同并发下的性能情况，各场景均达到性能标准，简述如下，具体参考各场景的详细测试报告：

1. 50并发登录的平均响应时间:1.35s,登出的平均响应时间：0.177s
2. 100并发登录的平均响应时间 2.29s，登出的平均响应时间0.295s
3. 200并发登录的平均响应时间5.2s，登出的平均响应时间0.358s
4. 300并发登录的平均响应时间 7.78s，登出的平均响应时间0.473s

**测试结果分析：**

随着并发量的增加登录响应时间呈线性增长，主要的影响因素为loadrunner服务器资源受限，在各个级别并发场景下各个应用服务器的cpu使用率保持在20%左右，各个mongodb服务的cpu使用率保持在20%左右。

在执行压力测试的同时，不断人工登录，通过监控工具检测的人工登录时间基本为500毫秒到800毫秒之间，少于1秒，登出时间为60毫秒到200毫秒之间，没有因为压力的加大而出现响应时间的波动。

基于mongodb的会话共享机制，对登录，登出的影响不大，可以在实际项目中推广使用。

## 7、应用集成

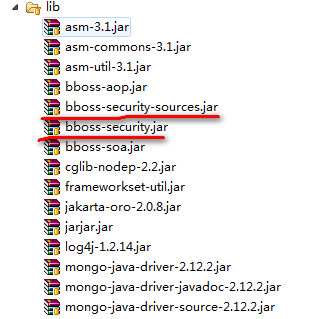
在J2ee项目集成会话共享，只需要将会话共享的相关jar和配置文件整合到项目工程中，然后进行相应的功能验证测试，可能需要编写序列化插件。

为了便于在项目中集成统一会话共享框架，准备了两个Demo工程：

1. 会话共享最小依赖jar和资源文件工程
2. 会话共享及监控最小依赖jar和资源文件工程

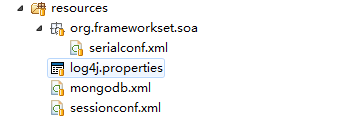
工程中包含以下资源：

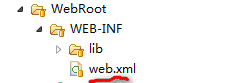
1. Session共享jar及依赖jar清单



1. Session共享配置资源文件

序列化插件配置文件，mongodb配置文件，session管理配置文件，Session过滤器配置文件





会话共享Demo使用可以参考文章：

《bboss会话共享demo使用指南》http://yin-bp.iteye.com/blog/2087308

集成共享框架过程中问题解决预案，**遗留系统集成会话共享组件出现以下情况需要对应用程序进行相应调整：**

1. 遗留系统修改存储在session中数据对象的属性和状态，但是没有重新set到session中

**修改预案：**调用session.setAttribute方法将修改后的对象再次存储到mongodb中的session表中，以便将修改共享给其他域应用或者集群应用节点。

1. 遗留系统存储在session中的对象数据无法采用bboss进行序列化或者反序列化

**修改预案：**调整对象定义或者为对象编写序列化插件，采取哪种方式视实际情况来定

1. 遗留系统定义并使用Session Event Listener，bboss提供了特有的session监听器

**修改预案：**需要将原来的session监听器迁移到bboss规范的session监听器

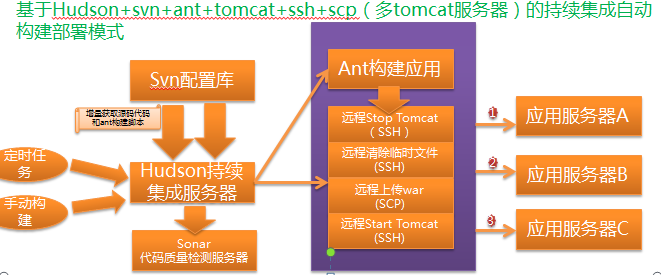
1. 遗留系统使用了重量级的session数据对象（对象中引用了很多大对象，这些大对象彼此没有关联，如果只需要获取或者修改其中的一部分数据，每次都要完整地将从mongodb中获取/写入这个完整的大对象，导致IO瓶颈和性能低下）

**修改预案：**将这些彼此独立的数据从大对象中剥离出来，作为独立属性存储到session中，从而避免不必要的开销。

上述预案都是实际使用bboss会话共享框架过程中总结出来的，可以不断补充这个预案清单。

## 8、应用部署

采用会话共享的应用系统一般都会采用集群模式进行部署，商业的应用服务器部署集群应用非常麻烦，代价也非常高昂（采购商业套件，部署成本高，难度大），我们经过实践摸索，形成了一套高效廉价部署方案：



**说明：**按先后顺序将应用部署到应用服务器，只停用正在部署应用的tomcat服务器，部署完毕后立即启动tomcat，再依次完成其他应用服务器应用部署，这样可以保证部署过程中其他服务器仍然处于工作状态，由于有会话共享机制的保障，前端用户使用系统基本不受影响，从而达到无停机部署和升级应用的目标。

## 9、监控统计

统一会话共享平台提供独立的在线会话管理和统计监控功能：

1. 查看应用在线用户数
2. 按条件查询应用会话信息
3. 应用会话管理

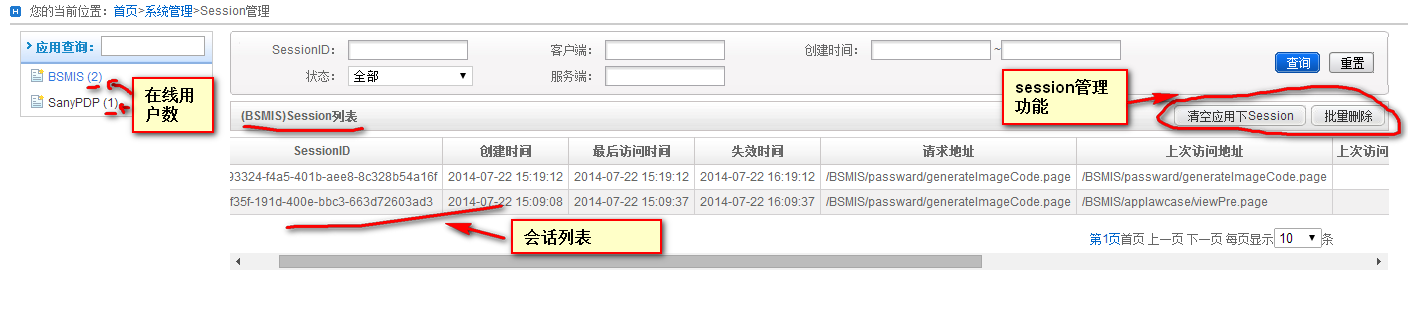
删除会话数据，包括删除特定应用的所有会话记录和删除一个或者多个用户会话数据，一旦用户会话数据被管理员删除，正在使用系统的用户会立即退出系统，只有重新登录后才能访问系统。

1. 查看会话明细

会话基本信息查看，共享会话数据查看，应用私有会话数据查看

1. 过期会话销毁扫描进程，定时销毁已经失效或者超时的session数据

监控界面效果示意图：



详细会话信息界面：

